

A decomposition-coordination method for the cascade interconnected dams problem

SOWG

October 10, 2017

Hydroelectricity is the main renewable energy which is produced in France (over 10% of the country annual production). It provides a clean (no greenhouse gases emissions) and fast-usable energy that is cheap and substitutable for the thermal one. We address the question of the production management of cascade-interconnected dams by means of a finite-horizon Markov decision process. Solving such a problem is not straightforward since it displays a large size, so is likely to suffer from the curse of dimensionality. We handle it with a decomposition-coordination algorithm that we adapt by approximation to the stochastic case.

Contents

1	Description of the problem	1
1.1	Dynamics of a single dam and dams cascade	1
1.2	The optimization problem	2
1.3	Numerical instance	3
2	Exact solution by dynamic programming	4
3	Approximate solution by dual approximate dynamic programming	5
3.1	Dualization of the coupling constraint and DADP	6
3.2	A heuristic to construct an admissible solution	9

Introduction

This practical work will follow the lecture entitled Decomposition-coordination methods in stochastic optimal control (P. Carpentier). **The computations will be performed using nsp and only nsp.** In order to launch nsp write `nsp &` in a terminal. Type `pwd` to see the current repertory and then download the practical work material from the Internet to this repertory. Since we will use a three dimension dynamic programming algorithm, a part of the programming functions had to be written in C. Moreover, we have to load some scenarios datas. Thus, we will need a specific nsp toolbox called `COURSES/stochdec`. In order to load it, go to the `stochdec` repertory and type `exec('loader.sce');`.

1 Description of the problem

1.1 Dynamics of a single dam and dams cascade

Let time t vary in $\{0, \dots, T\}$. The following real valued random variables are defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$:

- \mathbf{X}_t^i is the *storage level* at the beginning of period $[t, t + 1)$; (state)
- \mathbf{U}_t^i is the *hydropower outflow* during $[t, t + 1)$; (production control)
- \mathbf{Z}_t^i is the *inflow from dam $i - 1$ into dam i* during $[t, t + 1)$; (coordination control)
- \mathbf{W}_t^i is the *exogenous inflow* into dam i during $[t, t + 1)$; (noise)

The dynamics of the dam reads:

$$\mathbf{X}_{t+1}^i = \min \{ \mathbf{X}_t^i + \mathbf{W}_t^i + \mathbf{Z}_t^i - \mathbf{U}_t^i, \bar{x}^i \} \quad \text{with} \quad \mathbf{X}_0^i := x_0^i. \quad (1)$$

When $\mathbf{X}_t^i + \mathbf{W}_t^i + \mathbf{Z}_t^i - \mathbf{U}_t^i$ is greater than \bar{x}^i , the difference is spilled from dam i to dam $i + 1$ so that the upper bound constraint on dam i is directly handled by the dynamics. Therefore the total outflow from dam i to dam $i + 1$ is given by:

$$g_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_t^i, \mathbf{Z}_t^i) := \max \{ \mathbf{X}_t^i + \mathbf{W}_t^i + \mathbf{Z}_t^i - \bar{x}^i, \mathbf{U}_t^i \}.$$

The bounds constraints are:

$$\underline{x}^i \leq \mathbf{X}_{t+1}^i \quad \text{and} \quad 0 \leq \mathbf{U}_t^i \leq \min \{ \bar{u}_t^i, \mathbf{X}_t^i + \mathbf{W}_t^i + \mathbf{Z}_t^i - \underline{x}^i \}, \quad \forall t \in \{0, \dots, T - 1\}. \quad (2)$$

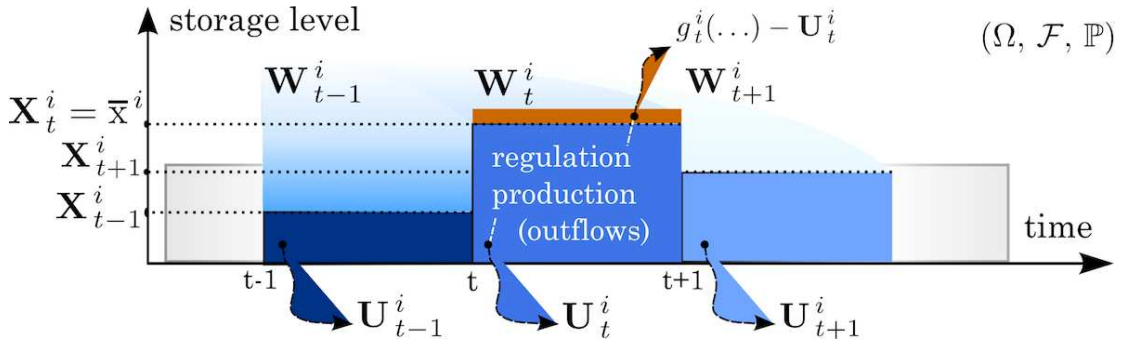


Figure 1: Dynamics of dam i

Each dam $i \in \{2, \dots, N\}$ receives from the dam $i - 1$ outflows $g_t^{i-1}(\mathbf{X}_t^{i-1}, \mathbf{U}_t^{i-1}, \mathbf{W}_t^{i-1}, \mathbf{Z}_t^{i-1})$ as inflows \mathbf{Z}_t^i and sends its own outflows into the dam $i + 1$ reservoir, at every $t \in \{0, \dots, T\}$. With the notation $\mathbf{Z}_t^1 \equiv 0$, this leads to the following coupling constraints:

$$\mathbf{Z}_t^i = g_t^{i-1}(\mathbf{X}_t^{i-1}, \mathbf{U}_t^{i-1}, \mathbf{W}_t^{i-1}, \mathbf{Z}_t^{i-1}) \quad \forall (i, t) \in \{2, \dots, N\} \times \{0, \dots, T - 1\}. \quad (3)$$

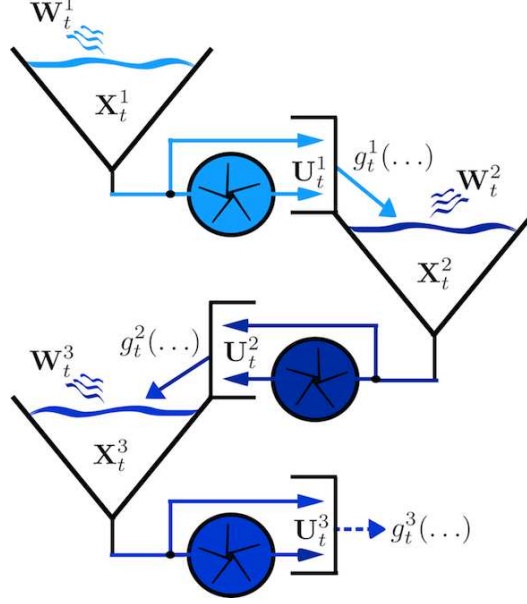


Figure 2: Dynamics of three cascade-interconnected dams

1.2 The optimization problem

We assume that the noise random variable $\mathbf{W}_t = (\mathbf{W}_t^1, \mathbf{W}_t^2, \mathbf{W}_t^3)$ are independent over time. The hydroelectric production obeys the following valorization mechanism for every dam:

- the production at time t , \mathbf{U}_t^i , is p_t linearly retributed and a quadratic cost to produce $\varepsilon \times \mathbf{U}_t^{i2}$ has to be considered,

$$L_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_t^i) := -p_t \times \mathbf{U}_t^i + \varepsilon \times \mathbf{U}_t^{i2};$$

- a final storage level appreciation function $K^i(\mathbf{X}_T^i)$ prevents the reservoir from being lower than a target storage level \hat{x}^i at the end of the study period,

$$K^i(\mathbf{X}_T^i) := a^i \times \max\{\hat{x}^i - \mathbf{X}_T^i, 0\}^2.$$

We consider the expected value of the sum of every dam total gain as the problem criterion:

$$\mathbb{E} \left(\sum_{i=1}^N \left(\sum_{t=0}^{T-1} -p_t \times \mathbf{U}_t^i + \varepsilon \times \mathbf{U}_t^{i2} + a^i \times \max\{\hat{x}^i - \mathbf{X}_T^i, 0\}^2 \right) \right). \quad (4)$$

Note that this criterion is both time and space separable. The control $\mathbf{U} = (\mathbf{U}_0^i, \dots, \mathbf{U}_{T-1}^i)_i$ must satisfy the non anticipativity constraint in the hazard-decision framework. So, \mathbf{U}_t^i has to be measurable with respect to the noises up to time t :

$$\mathbf{U}_t^i \preceq \mathcal{F}_t \quad \text{where} \quad \mathcal{F}_t := (\mathbf{X}_0, \mathbf{W}_0, \dots, \mathbf{W}_t) \quad \forall t \in \{0, \dots, T-1\}. \quad (5)$$

The problem consists in minimizing the criterion (4) under the dynamics (1), the coupling (3), the bounds (2) and the measurability (5) constraints.

1.3 Numerical instance

We will test algorithms on a valley composed of three cascade-interconnected dams over a $T = 12$ time steps period. The problem is of a discrete type and the dams share the following common characteristics:

- the volume \mathbf{X}_t^i of every dam i at t belongs to the forty one integers set $\{0, 2, \dots, 78, 80\}$;
- the space in which every control \mathbf{U}_t^i takes its values is discretized over the six integers set $\{0, 8, \dots, 32, 40\}$;
- the noise random variables \mathbf{W}_t^i are considered as being interstage independent and uniformly distributed over a finite set of ten integers.

The simulation part is based on 500 intakes scenarios.

Question 1 *Execute the script `answer-question1.sce` which is stored in the directory `COURSES/stochdec`. This script will draw N simulation scenarios for the dam number 1. Take a look at the script and change it to draw the scenarios of the other dams.*

The contents of the script `answer-question1.sce` is reproduced here:

```
// You should be in COURSES/stochdec to run this script
exec('loader.sce');
SIMSCEN=dp_get_simulation_scenarios();

// get the inflows law on dam 1
M=fscanfMat('Datas/Generated_datas/generated_intakes_0.txt');

// QUESTION 1:
xclear();
[nss,v]=size(SIMSCEN)
N=10
// Draw N scenarios of intakes of dam 1
Sim=SIMSCEN(1:N,:)+1;// take care to add one
T=dp_horizon();
res=Sim;

for t=1:T-1 do
    res(:,t)=M(Sim(:,t),t);
end;

xset('colormap',jetcolormap(N));
plot2d([],res');
xlabel(sprintf('first %d simulation scenarios',N));
```

2 Exact solution by dynamic programming

The noises W_t^i are independent over time, so that the problem can be theoretically solved by stochastic dynamic programming (DP). Although, DP is prone to face the curse of dimensionality (the method is not numerically tractable as soon as the number of dams is larger than 5) you will first solve the three cascade interconnected dams problem by DP since it is numerically possible (as $N = 3$). Note that the discretization of the problem avoids us to use an interpolation scheme to implement the method.

Question 2 *Running DP on the cascade of three connected dams is obtained using the function `dp_run()`. Then call the function `dp_simu()` as follows `[cost, stocks, controls]=dp_simu()` to obtain the expected cost and the stocks and the controls trajectories. Plot the stocks trajectories and show the optimal expected cost. Note that, for example, you will obtain the stocks evolution of dam 1 by using `S1=stocks{1}`.*

Question 3 *For the most sceptical of you, just try to beat the DP-computed strategy by building your own strategy ! ;) For example, you may want to multiply the optimal control mapping by a scalar $\gamma \in [0, 1]$ to get a new control U (you can use `dp_getu(U)` and `dp_setu(U)` to obtain or set the current used control table). Simulate the strategies obtained with your control strategy by using `dadp_simu()` and compare to what you had obtained before.*

Question 4 *You can also design your own strategy using function `dp_f2u` and simulate your strategy as seen in question3. How far is your solution expected cost from the optimal one?*

The next script gives some hints on how to use `dp_f2u`:

```
function u=myfeedback(t,w,x)
// scalar t, vector w of size 4 (three inflows and a price)
//, vector x of size 3 (water levels of the dams)
    u=sin(t)*x(1)*w(4)*[0;1;0]+x(3);
endfunction
// dp_f2u(fn): use a feedback function given as
// argument to fill a control array U.
```

```
U=dp_f2u(myfeedback)
```

```
...
```

3 Approximate solution by dual approximate dynamic programming

In a few words, we decompose the three dimension dynamic state problem into three one dimension dynamic state problems. This is done by:

1. dualizing the coupling constraints;
2. coordinating the local optimal solutions to get the global optimal one using an Uzawa algorithm.

3.1 Dualization of the coupling constraint and DADP

We aim at dualizing (3) and at solving the problem by using an Uzawa algorithm: at iteration k , the associated multiplier is a fixed \mathcal{F}_t -measurable random variable $(\lambda_t^{i+1})^{(k)}$, and the term (under the expectation) induced by duality in the cost function is

$$(\lambda_t^{i+1})^{(k)} \cdot \left(\mathbf{Z}_t^{i+1} - g_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_t^i, \mathbf{Z}_t^i) \right),$$

It can be decomposed as

- $(\lambda_t^{i+1})^{(k)} \cdot \mathbf{Z}_t^{i+1}$: term pertaining to dam $i + 1$.
- $-(\lambda_t^{i+1})^{(k)} \cdot g_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_t^i, \mathbf{Z}_t^i)$: term pertaining to dam i .

Finally, the following term is added to the cost of dam i

$$(\lambda_t^i)^{(k)} \cdot \mathbf{Z}_t^i - (\lambda_t^{i+1})^{(k)} \cdot g_t^i(\mathbf{X}_t^i, \mathbf{U}_t^i, \mathbf{W}_t^i, \mathbf{Z}_t^i).$$

The main idea of the price decomposition of the problem is to see \mathbf{Z}_t^i as an independent control variable as shown in figures 3. As \mathbf{U}_t^i does, \mathbf{Z}_t^i has to observe the (3) constraint. Then, we would like to solve the resulting N independent one-dam subproblems by DP¹,

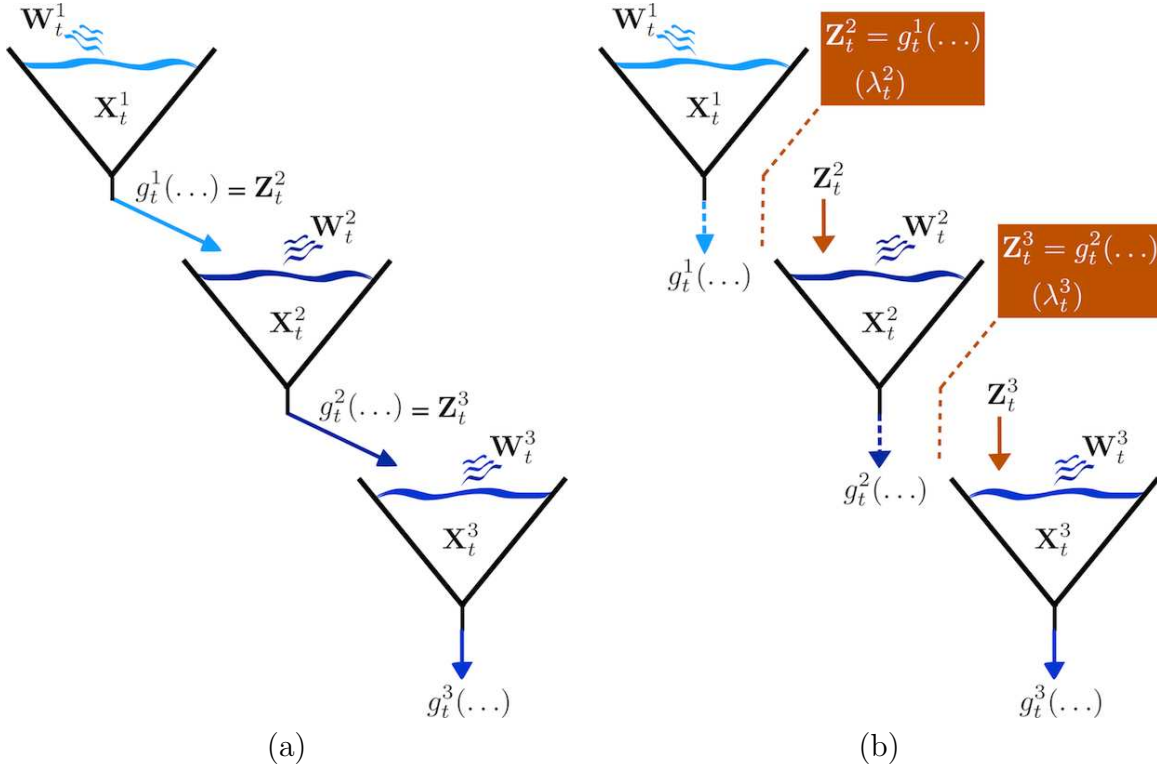


Figure 3: (a): Whole Problem (b): Decomposed Problem

$$\begin{aligned} \min_{\mathbf{U}^i, \mathbf{Z}^i, \mathbf{X}^i} \mathbb{E} \left(\sum_{t=0}^{T-1} \left(L_t^i(\dots) + (\boldsymbol{\lambda}_t^i)^{(k)} \cdot \mathbf{Z}_t^i - (\boldsymbol{\lambda}_t^{i+1})^{(k)} \cdot g_t^i(\dots) \right) + K^i(x_T^i) \right) \\ \text{s.t. } (1)_{|i}, (2)_{|i} \text{ and } (5)_{|i}. \end{aligned} \quad (6)$$

But, the presence of the random variables $(\boldsymbol{\lambda}_t^i)^{(k)}_{t=0, \dots, T-1}$ prevents to use dynamic programming in a straightforward manner since the random variables λ_t^i are not independent over time. The idea of DADP is to replace the multipliers $(\boldsymbol{\lambda}_t^i)^{(k)}$ by their conditional expectations w.r.t. chosen information variable \mathbf{Y}_t^i , namely

$$(\hat{\boldsymbol{\lambda}}_t^i)^{(k)} := \mathbb{E}((\boldsymbol{\lambda}_t^i)^{(k)} \mid \mathbf{Y}_t^i).$$

Note that this is equivalent to replace the (3) constraint by $\mathbb{E}(\mathbf{Z}_t^i - g_t^{i-1}(\dots) \mid \mathbf{Y}_t^i) = 0$. In practice, \mathbf{Y}_t^i is a short-memory process that will enter the state variables of the subproblems. In the sequel we will consider that $\mathbf{Y}_t^i \equiv \text{const}$, so that $(\hat{\boldsymbol{\lambda}}_t^i)^{(k)} := \mathbb{E}((\boldsymbol{\lambda}_t^i)^{(k)})$.

Thus, the Uzawa algorithm reads:

1. **initialization:** choose initial values for the multipliers $(\hat{\boldsymbol{\lambda}}_t^i)^{(0)}$ for all i and t ,
2. **minimization:** solve N independent one dimensional subproblems by dynamic programming,

$$\begin{aligned} \min_{\mathbf{U}^i, \mathbf{Z}^i, \mathbf{X}^i} \mathbb{E} \left(\sum_{t=0}^{T-1} \left(L_t^i(\dots) + (\boldsymbol{\lambda}_t^i)^{(k)} \cdot \mathbf{Z}_t^i - (\boldsymbol{\lambda}_t^{i+1})^{(k)} \cdot g_t^i(\dots) \right) + K^i(x_T^i) \right) \\ \text{s.t. } (1)_{|i}, (2)_{|i} \text{ and } (5)_{|i}. \end{aligned}$$

3. **coordination:** update the multipliers by a gradient step as follows:

simulation: apply the current strategy on a sample of n noise trajectories ω_l , to get n realizations of the random variables $(g_t^i(\dots))^{(k+1)}(\omega_l)$ by time step, compute all discrepancies $(\delta_t^i)(\omega_l)$ in the coupling constraints given by:

$$(\delta_t^i)(\omega_l) := \mathbf{Z}_t^i(\omega_l) - g_t^{i-1}(\mathbf{X}_t^{i-1}, \mathbf{U}_t^{i-1}, \mathbf{W}_t^{i-1}, \mathbf{Z}_t^{i-1})(\omega_l),$$

gradient step: update the multipliers by taking the average of the discrepancies $(\delta_t^i)(\omega_l)$

$$(\hat{\boldsymbol{\lambda}}_t^{i+1})^{(k+1)} = (\hat{\boldsymbol{\lambda}}_t^{i+1})^{(k)} + \frac{\rho_t^i}{n} \times \sum_{l=1}^n \delta_t^i(\omega_l)$$

¹These subproblems are one dimensional problems and could have been numerically solved this way if the multipliers random variables would have been known.

4. **stopping condition:** go to step (2) until convergence of the algorithm.

Question 5 Run the script `job.sce` to see iterations of the DADP algorithm solving the three cascade interconnected dams problem. You should observe the multipliers expected values and the dual costs converging in about three thousand iterations. At the end of iterations you can obtain the optimal multipliers expected values using the function `dadp_get_lambda()`. Use this function to store this values in matrix `OPT_LAMBDA` by calling `OPT_LAMBDA=dadp_get_lambda()`.

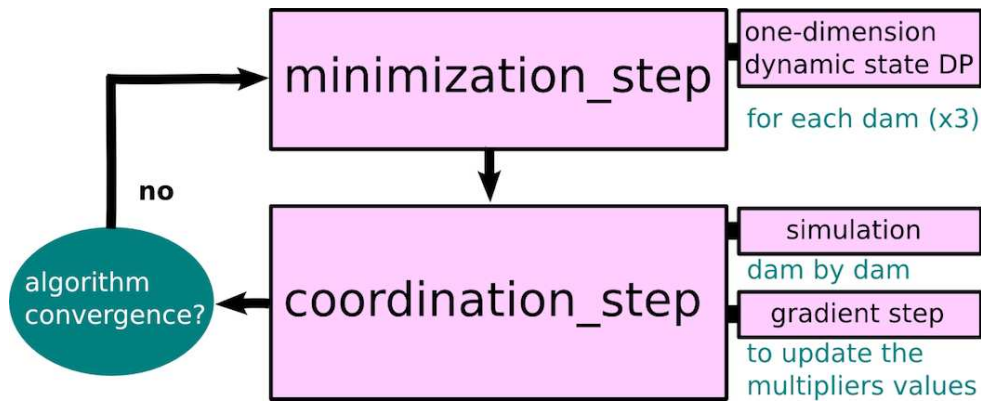


Figure 4: Uzawa algorithm

Question 6 Have a look at the Uzawa algorithm figure Figure 4. Implement your own gradient step process and make the gradient step size values vary to see how it affects the convergence of the algorithm. You can use the following program skeleton.

```

exec loader.sce
nspoc('load_data');
l1=[];l2=[];cost=[],costa=[],costaa=[],costab=[];
dadp_set_gstep([1*ones(11,1),0.1*ones(11,1)]);
for i=1:200
    dadp_minimization_step(); // DP for each dam given lambdas
    // now you should update the lambdas using the following functions
    // dadp_coordination(): computes the lambda gradients that you can then
    // obtain with dadp_get_dlambdas();
    // then dadp_get_gstepsize() returns the gradient stepsizes.
    // you can get or set the current values of lambda with
    // L=dadp_get_lambda();
    // or
    // dadp_set_lambda(L);
end
  
```

Question 7 Write your own `dadp_coordination()` function using the following code skeleton and then compare your code to the reference one.


```

function [dl,dc]=dadp_coordination1()
    ndams = dadp_ndams();
    T = dadp_horizon();
    Nu = dadp_n_uzawa_scenarios();
    // dc: used to store the dual cost
    // dl: used to accumulate the average gradients it should be of size
    // zeros(2,T-1);
    for ksimul=0:Nu-1
        // dadp_x0(); // gives the initial stocks
        for t=0:T-2
            // dadp_get_noise(ksimul,t); // get the intakes and price
            // [xp,u,g,costs]=dadp_dynamics(ksimul,t,x,w);
            ...
        end
        // dadp_final_cost(x); // returns the final cost for each dam
    end
    dc =- dc /Nu; // average
    dl = dl/Nu; // average
endfunction

```

Question 8 *You can change the previous function dadp_coordination1 in order to get the deviations from the coupling constraints trajectories at the optimal multipliers expected values. Compute these values when at the optimal multipliers expected values that you have stored before.*

The approximation $\mathbf{Y}_t^i \equiv \text{const}$ is a relaxation of the problem (as the constraint is loosened). Thus, the strategies that we derive may not be feasible, even if the algorithm converges: we have to design a method to go back to a feasible strategy.

3.2 A heuristic to construct an admissible solution

Once the algorithm has converged we have feedbacks laws that verify the constraint

$$\mathbb{E}\left(\mathbf{Z}_t^i - g_t^{i-1}(\mathbf{X}_t^{i-1}, \mathbf{U}_t^{i-1}, \mathbf{W}_t^{i-1}, \mathbf{Z}_t^{i-1})\right) = 0.$$

But, this means that the physical constraint $\mathbf{Z}_t^i = g_t^{i-1}(\dots)$ *a.s.* is not verified. Consequently one has to define an heuristic to turn this strategies into an admissible one.

We first note that the optimal control to apply along a given noise trajectory may be computed using the Bellman function. The way to do this is the following: suppose that you have the three dimension DP-computed Bellman function

For every scenario

For t=0:1:T-1

```

get the current state and the inflow value for every dam;
set a variable opt_cost to infinity;
For u_i=0:4:40 (a loop a dam, three loops here thus)
    set the corresponding instant cost to a variable cost;
    apply the dynamic to get the reached states;
    add the reached states t+1 Bellman function value to cost;
    If cost < opt_cost then
        opt_cost=cost and set the u_i values to a vector opt_u;
    apply the dynamics with u=opt_u and collect the new current states;

```

Question 9 Use the simulation part of the previous coordination function to implement the simulation using the Bellman function you get by DP and compare it to the `dp_simu()` results.

We don't have the three dimension Bellman function but we can construct an approximate Bellman's value function for the global problem as the sum of the Bellman's value function of each subproblem. Consequently we obtain a global admissible strategy by doing a one time step optimization of the global problem. More precisely if we write $V_t^i(x_t^i)$ the Bellman's function obtained for subproblem i when the DADP algorithm has converged, we define $\tilde{V}_t(x_t) = \sum_{i=1}^N V_t^i(x_t^{i-1}, x_t^i, x_t^{i+1})$, with the convention that $x_0^t = 0$. Then the control we choose at time t when the cascade of dams is in the state x_t and the w_t is observed is the optimum of

$$\min_{u_t} \sum_{i=1}^N L_t^i(x_t^i, u_t^i, w_t^i) + \tilde{V}_{t+1}(f(x_t, u_t, w_t)) .$$

This problem is global on the cascade but only done on one time-step, so numerically tractable.

Question 10 Apply the heuristic to the optimal strategy we computed using `dadp_admissible()` and simulate with `[cost, stocks, controls]=dadp_simv()`. Use the function which checked for the feasibility of the DADP computed strategy to see that the strategy is now feasible. Compare the results with those we get by the application of the exact optimal strategy.